Detection and mitigation technique against collusion attack in RPL-Based IoT networks

DAVID ESSAADI
Supervised by:
CHHAGAN LAL , MAURO CONTI
TU Delft

Abstract

The Internet of Things (IoT) is a technology used in applications varying from home- and industrial automation to medical devices, smart vehicles, fitness trackers and many more. Such IoT networks often consist of incredibly resource-constrained devices, and are known as Low-power and Lossy Networks (LLNs). The Routing Protocol for LLNs (RPL) aims to provide a routing standard for such networks. Due to the tough performance constraints, RPL is unable to provide strong security guarantees. Many researchers are posing attacks against RPL-based networks, and with the increasing number of implementing devices, it is important that research is done to ensure message integrity and network reliability. In this paper we concern ourselves specifically with collusion attacks. We propose Hop-Count Reachability (HCR), a mitigating method against the coordinated blackhole attack. In HCR, leaf nodes periodically ping the root node with DAO messages. If the root node is reachable, it will reply with a DAO-ACK, upon which the leaf node sleeps for a period of time. When the number of missed ACKs in a certain time frame exceeds a certain threshold, the affected node may identify the attack and mitigate it by selecting a new parent. HCR may increase control packet overhead anywhere between 1.6 and 25% depending on the chosen parameters, and successfully mitigates the coordinated blackhole attack in all scenarios where affected nodes can choose an (eventually) unaffected parent.

1 Introduction

The Internet-of-things (IoT) has become increasingly popular. IoT devices measure, process information or interact with the real world and cooperate with each other without human interaction to improve our daily lives [1]. The IoT has many different applications, ranging from smart-home solutions like smart lighting, automatic temperature control and voice assistants to traffic lights, smart vehicles and even medical equipment. Additionally, the IoT is used to enable industrial automation and in automating emergency decision making [1]. The demand for IoT devices is increasing and expected to continue growing in the future. However, this new technology does not come without its drawbacks. The IoT needs to support routing messages for networks with enormous quantities and variety of devices, with need for strong horizontal integration. Moreover, many IoT devices are limited in processing power, memory and battery-life [6]. Networks consisting of such limited performance devices, connected by lossy links, are known as Low Power and Lossy Networks (LLNs). Due to the variety of supported devices and their performance limitations, it is infeasible to use the traditional internet routing protocol for LLNs [1].

To tackle the routing problem for limited performance devices, the Routing Protocol for LLNs (RPL) has been designed and standardized in RFC6550 in 2012 [6]. RPL aims to enable IPv6 connectivity for LLNs by allowing low-performance wireless devices to communicate over one or more hops to deliver their data to other IoT devices either directly (P2P) or by communicating with a single link(MP2P / P2MP) [6]. Due to the performance limitations RPL needs to adhere to, the protocol is unable to support strong security mechanisms like TCP. The lack of strong built-in security, the wide adoption of RPL and the immaturity of the protocol are reasons for researchers to constantly propose new attacks against RPL. As of 2020, there are ten generic attack types against IoT networks, consisting of four hole attacks (black-, worm-, sink-, gray-), HELLO flooding, clone, misdirection, network partitioning, routing loop and rushing attack [3]. Aside from the generic IoT attacks, five RPL-specific vulnerabilities exist. These are the local-repair, rank, DODAG version, DIS and neighbor attack [3]. In practice, combinations of attacks may be used to avoid detection or to increase the impact of the attack on the network.

RPL is a relatively new protocol and researchers are actively proposing attacks against it, while detection research is still in early stages. The great difficulty in securing RPL is that solutions must be as simple as possible in order to maintain a low performance footprint [3]. Related work concerns itself firstly with investigating RPL's security against non-colluding attacks like the Version Number Attack (VNA), for which a detection method is posed in [2]. Secondly,

related research has been performed on mitigating solutions against collusion attacks like the wormhole attack [5] and binary blackhole attack [9], which both require at least two colluding attackers to perform. The mitigation method against the binary blackhole attack, called Cuckoo-RPL, in [9] is especially interesting for us, as we are discussing the coordinated blackhole attack in this paper. Unfortunately, no performance data is available. This information would be beneficial in comparing the benefits between Cuckoo-RPL and the detection method posed in this paper. Research specific to collusion attacks in RPL is lacking and few mitigating solutions exist.

This paper focuses on the coordinated blackhole attack. The coordinate blackhole attack is particularly dangerous, since it is able to effectively disconnect large sections of an RPL-based network. Related work about coordinated routing attacks is considered to help present a novel mitigation technique against the attack. A theoretical analysis on metrics like control packet overhead and attack detection rate is performed based on a sample scenario. Finally, a discussion on the strengths and limitations of the proposal is given. The discussion is concerned with trade-offs in performance in certain situations and with other attacks the defense mechanism may mitigate. In this paper the following research question is answered: *Identify the state-of-the-art detection and mitigation solutions proposed specifically for coordinated routing attacks in RPL-based IoT networks. What limitations do these solutions exhibit? What are possible approaches that could be used to address one or more of these limitations?*.

In section 2 background information on RPL, constraints on RPL-based IoT networks and related work on collusion attacks and their known detection methods is presented. Next, we discuss the methodology of our proposal in section 3. Here we explain the coordinated blackhole attack and present an example scenario via which we describe how our mitigation technique works and how it mitigates the attack. We present an analysis on the impact the mitigation method has on the network, and discuss its strengths and weaknesses. In section 5, a summary of the methodology is given, along with a discussion of the importance of addressing collusion attacks. This section also contains a discussion on limitations and future research directions. Finally, section 6 concludes the paper.

2 Background and Related Work

First, a brief overview of the RPL protocol is given. Secondly, we discuss the constraints placed on RPL-based IoT networks. Finally, an overview of existing collusion attacks on RPL-based IoT networks is presented, to give an idea of the form in attackers may attempt to break the protocol. An existing detection/mitigation method is mentioned for every attack, along with a brief discussion on its strengths and limitations.

2.1 The RPL protocol

RPL is the the routing protocol designed for LLNs. RPL has been standardized in RFC6550 [8], with the aim to support routing in low-power and lossy IoT networks, as the regular internet protocols are unsuitable for such networks due to their inherent performance limitations. The idea behind RPL is that nodes in the network are working together to communicate with each other over one or more hops. In such a network, each node contributes to the routing structure. At the border between the regular internet and the RPL-based network is a special node, called a root node. Such a root node is able to communicate with both the regular internet and the RPL network. The root node is usually less resource constrained than other nodes in the network. Fundamentally, RPL uses a graph-like structure called a Destination Oriented Directed Acyclic Graph (DODAG) to enable routing. A DODAG is a special type of Directed Acyclic Graph (DAG). Packets may travel upwards (towards the root node), or downwards (away from the root node) along the DODAG. To be able to build the DODAG, all nodes in the network have a rank value associated with them; rank strictly increases in downward (away from the root node) direction. The exact rank value is determined by every node's Objective Function (OF). Nodes decide which OF to use depending on which network metrics need optimizing. Apart from being used for rank calculation, the OF is also used for parent selection, thereby being defining in the DODAG building process.

To enable communication, three control message types are used. A DODAG Information Object (DIO) is used to communicate information about an RPL instance; therefore DIOs play a vital role in the parent selection process, and maintaining the DODAG structure [8]. Destination Advertisement Objects (DAO) are propagated by a node upward along the DODAG to inform upper parts of the DODAG how to route toward lower nodes. Optionally, a DAO-ACK message may be sent in return, to inform the original sender that the DAO has been received. DODAG Information Solicitation (DIS) messages are used to request a DIO from any RPL node. A DIS is used to discover close RPL nodes, and is therefore usually sent as multicast to close neighbors. Apart from these control messages, RPL nodes send data packets to communicate the actual application data.

2.2 Constraints on RPL-based IoT networks

RPL was designed to support a network consisting of nodes with limited performance and lossy links. Ideally, this would mean any constrained device would be able to partake in the network. However some minimum constraints are put on RPL networks. A summary of these constraints can be found in the survey by H.Kim, J.Ko, D.Culler et al. in [6]. The main constraints we are concerned with are the:

- Resource constraint - RPL based networks must support as low as 8-bit devices limited to 128kB or 256kB of memory. The RPL should not use more than 1% of duty-cycle and/or provide a minimum of 5 years of battery lifetime [6].
- Node property awareness - RPL based networks must

take specific node characteristics like power availability and memory into account when deciding how to route a packet. [6].

- Security - RPL based networks must prevent attackers from participating in routing decision process to ensure message integrity [6].

We will define attackers in the network as nodes participating in the network while purposely violating one of these constraints, or forcing other nodes to violate these constraints.

2.3 Related work

Collusion attacks are attacks in which two or more attackers need to cooperate to perform the attack. Some types of collusion attacks are the wormhole attack, coordinated blackhole attack and Distributed Denial-of-Service (DDoS) attacks. An overview of these attacks and relevant research is given here.

The *wormhole attack* is an attack which uses two or more attackers to build a tunnel between them. These tunnels are used to push traffic from one end of the tunnel to the other end through a faster, reliable link. The tunnel allows distant malicious nodes to attract traffic by advertising as close neighbors. Whenever one end of the tunnel receives a packet, it gets forwarded quickly to the other side of the tunnel, which then replays the packet. When the original packet finally reaches the destination, it arrives later than the replayed packet and is therefore discarded by the receiver. Wormhole attacks are notoriously difficult to detect due to their ability to communicate over a single hop via reliable links. Some mitigation methods against the wormhole attack use packet leashes, location-based keys or more expensive hardware in the form of directional antennas [3]. Another solution uses centralized computing to keep track of routing topology. A more invasive method called DAWWSEN mitigates the wormhole attack by creating a hierarchical DODAG to use for routing [3]. The internal working of these mitigation methods is out of scope for this paper.

The *binary blackhole attack* is a colluding adaptation of the blackhole attack. In the regular blackhole attack, a single attacking node advertises a misleadingly low rank in its DIO messages to get other nodes to choose it as their parent. Whenever the attacking node receives a packet from one of its children, it drops it, resulting in severely reduced network performance. The watchdog mitigation technique can be used to detect the blackhole attack. Watchdog nodes keep track of how many of their transmitted packets their parent node forwards to its successor. If too many unsuccessful forwards are detected, the malicious parent node is blacklisted and a new parent is selected. The binary blackhole attack avoids this detection method by using multiple malicious nodes. Using two closely located malicious nodes A and B, the watchdog method can be fooled [9]. Like in the regular blackhole attack, node A advertises a lower rank to attract neighbors. The difference is that node A does not drop the packets, but forwards packets to its companion node B. Node B is acting like a seemingly regular node to the rest of the network, but whenever B receives a forwarded packet from node A it drops the packet. Because A forwards the packet to another node, the children of node A believe their packet is properly forwarded, thereby node A is avoiding the Watchdog method. However, when B drops the packet, A and B have succeeded in performing the blackhole attack together, with detrimental network performance as a result.

Zhang et al. propose the Cuckoo-RPL mitigation method against the binary blackhole attack [9]. In Cuckoo-RPL, a data structure called a Cuckoo Filter [4] is used to add authentication to the RPL. The Cuckoo Filter is a special hash-table, which is meant to provide high space efficiency. The hash-table is used to store the fingerprints of all legal nodes in the network. In Cuckoo-RPL, all nodes keep track of their own Cuckoo Filter. To be able to participate as parents in the network, a node first needs to be registered with the root node to obtain its share key. This is done via a secure transmission method. Once registered, the nodes can be placed in their correct locations, after which the root node begins the DODAG building process. Initially, the only entry in every node's hash-table is the fingerprint of the root node. All nodes in the network send unicast DAO messages towards the root node, which in turn computes their hash based on their IP address and adds it to its hash-table. The root node may send a DIO with the UpGrade (UG) flag set to let every node compare their hash table with the one provided by the root node and update it if needed. To ensure the provided hash-table is actually sent by the root node, it signs it with its key, which the other nodes can then verify. Whenever a DIO message is received, nodes first do a lookup in their hash-table to find out whether the sender is a legal node in the network. If the node is registered, the DIO message is processed, whereas unregistered nodes' messages will be ignored.

The *Distributed Version Number Attack* (DVNA) is an attack in which multiple malicious nodes initiate illegal global repairs. In the RPL specification, the assumption is made that only the DODAG root can initiate a global repair. In practice however, any node can modify its messages' version number, triggering close nodes to initiate a global repair. This attack is called a Version Number Attack (VNA). The VNA is a strong Denial of Service (DoS) attack against RPL, reducing packet delivery by up to 50 %, increasing latency by up to 6 times, and increasing power consumption by up to three times [2].

Arış, et al. propose a detection method called *shield* to guard against this attack [2] . In shield, a trust-based change in DODAG repair is proposed, in which a node only triggers a global repair when the majority of its close upward neighbors increase their version numbers. While this detection method would work for a single attacker it may fail when multiple attackers initiate a VNA at the same time. This is caused by the assumption that information provided by most nodes is truthful, where in practice it may very well be the case that multiple attackers are colluding. When multiple attackers are able to cooperate close together, they can essentially perform

a DDoS on all of their children nodes.

3 Methodology

The following approach is taken to answer the research question:

- A description of the coordinated blackhole attack is given along with a diagram of how it affects the network.
- A brief discussion of the strengths and limitations of the existing mitigation method against the binary blackhole attack is given.
- A proposal for a novel mitigation method - called HCR - is presented.
- The performance of HCR is analysed along multiple metrics.
- The limitations of HCR are addressed in section 5.

3.1 Coordinated blackhole attack

The coordinated blackhole attack is an adaptation of the regular blackhole attack. In the blackhole attack a malicious node simply drops all packets it receives. Often, the blackhole attack is used in combination with a sinkhole attack, making the impact on the network more severe. However, a regular blackhole attack is relatively easy to detect and several papers have proposed defense mechanisms against it. *Zhang et al.* propose the binary blackhole attack as a way for the blackhole attack to avoid detection by the watchdog method [9]. In the binary blackhole attack, two nodes (A and B) are working together to avoid detection. They do so by having malicious node A forwarding packets to malicious node B, which performs the actual dropping of packets. To the outside world, it seems like A is properly forwarding packets to B, therefore legal nodes will not decide to block node A from the network. This attack is detrimental to network performance, as it effectively blocks out any children of A from partaking in the network. The binary blackhole attack always uses exactly two malicious nodes to perform the attack. The coordinated blackhole attack is an adaptation of this in which any number of malicious nodes may be chained together to perform the attack. This chaining of nodes may become increasingly detrimental to network performance when used in combination with a coordinated sinkhole attack. In this scenario most nodes will choose the malicious node as their parent under the assumption packets are arriving correctly, when in fact all children of the chain of malicious nodes are unable to participate in the network.

3.2 Strengths and limitations of Cuckoo-RPL

In section 2 the workings of Cuckoo-RPL are discussed. It is a mitigation method against the binary blackhole attack, which uses node root registration and requires all legal nodes to maintain a hash-table of other legal nodes in the network. The benefit of Cuckoo-RPL lies in the assumption that a binary blackhole attack is only initiated after the network has been built. Under this assumption, attacking nodes are seen as new nodes in the network. Therefore attackers are expected not to be in each node's Cuckoo Filter - meaning attacker's DIO messages are ignored. However, in practice, more sophisticated attackers may first partake regularly in the registration process, to only launch their attack after the network has already been built - completely avoiding this mitigation method. Moreover, while the Cuckoo-Filter is meant to be light on memory usage, the memory footprint may still be significant in networks with thousands of nodes. On top of that Cuckoo-RPL only allows leaf nodes to be added dynamically to the network. Nodes would first need to register with the root node to be able to act as parents. Due to these limitations, a new detection method against the coordinated blackhole attack is presented in this paper.

3.3 Mitigating the coordinated blackhole attack

The goal of this paper is to design a mitigation method to the coordinated blackhole attack. To this end, an analysis of the different types of defense mechanisms is performed. Cuckoo-RPL is an authentication based defense mechanism. In such a system there are two types of nodes; authenticated and unauthenticated. Authenticated nodes are able to act as parents to other nodes, whereas unauthenticated nodes are only able to act as leaves in the network. This method has the benefit of working well for static networks, but the drawback of lacking dynamic capabilities. Moreover, nodes that are seen as legal nodes may in fact be compromised and attack the network from within. A different defense mechanism is a detection based mechanism like the Watchdog method. This defense mechanism has the benefit of working in both static and dynamic networks, but is limited by its performance and control message overhead. In addition, the Watchdog method may easily be circumvented by colluding attackers. Multiple Watchdog nodes would need to work to work together to detect colluding attacks, resulting in even more control message overhead and reduced network performance. Ideally, a system which allows for an acknowledgement of every received message is used, however this also increases control message overhead beyond reasonable levels. To this end, I propose a defense mechanism in which nodes periodically check whether they can reach the root node through their parent. It will be referred to as hop-count reachability (HCR).

In HCR the assumption is made that the network is running in non-storing mode. In non-storing mode, the root node is aware of the routing topology to the extent that it is able to address every known node in the network via one of its child nodes. Leaf nodes will periodically send a unicast DAO message directed toward the root node, with the 'K' flag set to true. By setting the 'K' flag, an explicit request for a DAO-ACK is made. In paths without attackers, the DAO should arrive at the root node, which then returns a DAO-ACK along the reverse path. In paths where the coordinated blackhole attack is performed, the DAO will never reach the root node, therefore no DAO-ACK will be received by the excluded node. If this happens, the leaf node will send multicast DIS messages to its neighbors, in the hope to locate another possible parent. If one is in range, the excluded node will connect to this parent instead and perform another HCR on this path. To ensure that regular packet loss does not immediately trigger a re-selection of parents, a counter

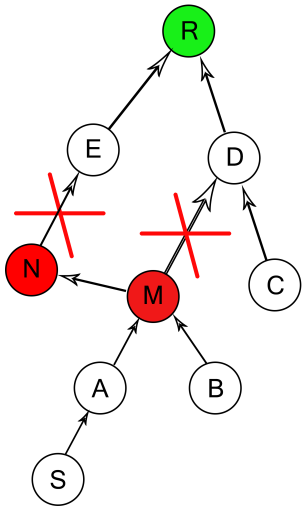

Figure 1: An example coordinated blackhole attacking scenario. The red nodes (M & N) are malicious nodes. The green node (R) is the root node. M acts as a regular node to outsiders, but always forwards packets to N, which in turn drops all packets it receives from M. It is assumed that all nodes are able to reach the nodes directly neighbouring them.

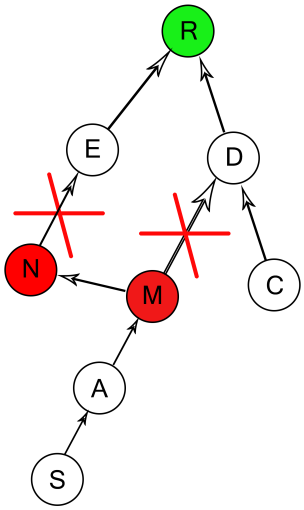

Figure 2: A slightly modified scenario of fig.1. In this scenario, the B node is missing. This causes the detection method to be unsuccessful in also mitigating the attack.

is used to check how many DAO-ACK messages are missed within a certain time frame. If a node counts more than n missed ACKs within this time frame, it will switch parents. In future extensions the node may inform the root node which node is misbehaving, to block malicious nodes from the network. HCR starts in leaf nodes, as paths from leaf nodes must certainly reach a possible attacker on the path to reach the root node. If a leaf node is unable to switch parents, it may request its current parent to perform HCR instead. A DAO with any of the 6 free flags set may be used for this. It is out of scope of this paper which flag is to be used exactly.

The attacking scenario of a coordinated blackhole attack is shown in fig. 1. Because M has been chosen as a parent by both nodes A and B, they are now unable to reach the root node and are unable to participate in the network. One could imagine nodes A and B as sensors, providing critical information to the root node in a regular situation. Now, however, this critical information is dropped and the root node does not receive the information it needs. In HCR, the leaf nodes (S, B, C in fig. 1), periodically send a DAO to the root node. In this case, node C is able to reach the root node, and will receive a DAO-ACK. However, nodes S and B are unable to reach the root node, and will try multiple times before deciding that an attack is being performed somewhere in the chain of their parents. In this scenario, node B will start looking for a new parent. It sends out multicast DIS messages, after which it will find nodes A and C as possible parents. Depending on its OF, B may either choose C or A. Note that it is not important in which order B selects a parent, as it will perform another HCR test as soon as it connects to its new parent. B will eventually select C as its parent and will be able to reach the root node through it. Node S is unable to select B as its parent because it is out of range. S will therefore request that node A performs HCR. Note that node A is now also able to connect to the network through node B by following the same process. Also note that it may be possible that a node switches between two parents that are both affected by the attack, even if an unaffected parent is available. To prevent this, a node performing HCR may connect to parents in order of increasing IP address, wrapping around.

3.4 Performance analysis of HCR

HCR is analysed according to the following metrics to gain a better understanding of its performance:

- *Packet Delivery Ratio (PDR)* - Measured as the percentage of packets that is successfully transmitted to the next node. A standard PDR of 99% per hop is assumed, which is reasonable for best-parent routing according to *J Tripathi et al.* [7]. HCR increases the number of packets sent in the network, which may increase the chance of collisions between simultaneously transmitting nodes. From theoretical analysis it is difficult to place an exact measurement on the decrease of PDR. Assuming a worst case of 10% increase in packet collisions, a 90% PDR is assumed in the rest of this analysis. A 90% PDR translates to a 59% chance of successfully transmitting a packet from root to leaf (or vice-versa) over 5 hops.
- *Attack Detection Ratio (ADR)* - Measured as the ratio of affected nodes detecting the coordinated blackhole attack. A theoretical analysis of the attacking scenario in fig. 1 is performed. The affected nodes in this scenario

are A, B and S - of which S and B are leaf nodes. Both B and S will detect the blackhole attack, since they will not receive any ACK packets from the root node. In general, all leaf nodes detect the attack, since they are guaranteed to reach a possible attacker on the path to the root node. Whenever the leaf node is unable to switch parents (S, in the scenario), it requests its parent (A) to perform HCR via a specialized control packet. The parent will then also detect the attack. In general all affected nodes will detect the coordinated blackhole attack, resulting in 100% ADR for children of the malicious nodes.

- *False Positive Rate (FPR)* - Measured as the probability of a node incorrectly assuming it is being attacked. A 90% PDR, hop-distance of 5 between root and leaf and threshold of 10 missed ACKs out of 10 per 10 minutes is assumed. The probability of a false positive is then equal to the probability that all ping/ACKs are unintentionally dropped. This translates to a FPR of 1.57% per 10 minutes. Higher thresholds will lower the FPR, whereas lower thresholds will increase it. Note that FPR is also dependent on the PDR. If PDR is significantly lower than 90%, the threshold needs to be increased.
- *Attack Mitigation Ratio (AMR)* - Measured as the ratio of affected nodes mitigating the coordinated blackhole attack. An analysis of the attacking scenario in fig. 1 is compared to the scenario in fig. 2. A 100% ADR is assumed. In scenario 1, B is able to switch directly to C as its parent - mitigating the attack for itself. S is unable to switch parents, therefore broadcasts a specialized control message to A to request A performs HCR. Node A detects the attack and switches to B as its parent. In this scenario the AMR appears to be 100%. However, as illustrated in scenario 2, when affected nodes are unable to switch parents to (eventually) unaffected nodes, they are unable to mitigate the attack. In general we say whenever nodes are able to switch parents to eventually unaffected nodes there is 100% mitigation. Any nodes unable to do so will have 0% mitigation. In most real-life scenarios it may be reasonable to assume that nodes are able to select a different parent.
- *Attack Detection Time (ADT)* - Measured in the average number of milliseconds (ms) it takes for an affected node to detect the coordinated blackhole attack. HCR is flexible in the number of pings per time-frame, which affects the ADT. We assume a delay of 5ms per hop for transmission of a ping packet and a distance of 5 hops between root and leaf. Assuming no delay is induced in calculating a response packet, it takes 50ms Round-Trip-Time (RTT) for a leaf to ping/pong the root node. The RTT is often negligible, except for situations in which a very high threshold and low sleep duration are chosen. In general the detection time is

$$threshold * sleep(ms)$$

  for the chosen time frame.
- *Control Message Overhead (CMO)* - Measured as the percentage of control packets transmitted compared to the total number of packets sent by a node. The performance evaluation in RFC6687 is used a base case to compare against. This research is performed on a DODAG of 45 nodes, optimized by link ETX [7]. Every node generates a data packet every 10 seconds. In table 1, an overview of CMO in such a network is presented. Observe that the root and its close nodes are used in much of the data packet forwarding, and CMO is almost negligible at as low as 3% overhead for these nodes. Leaf nodes do not need to route other nodes' data packets, and therefore have a relatively higher CMO of around 26%. Assuming HCR performs 10 pings within a 10-minute period, it increases the number of control packets transmitted from 600 to 610 per 10 minutes. This translates to a 1.6% increase in CMO for the leaf node, to a total of 26.7% CMO. More aggressive detection may be used. For instance bursts of 5 root-checks over a period of 20 seconds may be sent. This results in a significant increase of 150 control packets per time-frame, resulting in a total of 750 control packets generated by leaf nodes every 10 minutes. This translates to a 25% increase in CMO, to a total of 28.8% CMO. A discussion of trade-offs and limitations is given in section 5.

CMO in RPL according to scenario in RFC6687			
Hop-count	Control packets	Data packets	ratio
0 (root)	6e2	2e5	3%
1	9e2	2e4	5%
2	9e2	8e3	11%
3 (leaf)	7e2	3e3	23%

Table 1: Hop-count is the number of hops the observed node is away from the root node. Control packets and data packets are mean number of packets transmitted by the given node per 10 minutes (in scientific notation). Ratio is the ratio between control packets and data packets, in percentages.

4 Responsible Research

The intention of this research is to gain insight in the workings of colluding attacks against RPL-based IoT networks and to build foundations on which RPL may become a more stable and secure protocol.. By openly researching these attacks, more research can be done on the detection and mitigation of these attacks. This research is in no way intended to be used in unauthorized attacking of RPL-based networks, but is meant to give insight in the weaknesses of RPL and how to defend it against collusion attacks. By following the steps in the methodology, any researcher with knowledge on the subject can verify the proposal. This research is based on related literature and is backed by sources from IEEE, CCC and various RFCs. Theoretical analysis was used in this paper since time did not allow for simulation testing of HCR. This may be addressed in future work to gain more insight into the performance and feasibility of HCR.

5 Discussion

In this section a summary of the methodology is given. The importance of addressing coordinated attacks, and specifically the coordinated blackhole attack is discussed. Furthermore, a discussion on the performance and limitations of HCR along the metrics in section 3. Finally, some future research directions against collusion attacks in RPL are given.

5.1 Main points of methodology

In section 3 the coordinated blackhole attack, Cuckoo-RPL and HCR are proposed. HCR works by letting leaf nodes periodically send DAO messages up towards the root node, with the 'K' flag set. By setting this flag, the root node will respond with a DAO-ACK message back to the leaf node. This method works like a form of ping, ensuring that the leaf node is still connected to the DODAG properly. If *n* root-checks are done without a successful ACK within a certain time frame t, the leaf node will assume that an attacker is performing the blackhole attack in its chain of parents, and will look for a different parent itself. This results in one of two options:

- The leaf node finds a new parent, thereby possibly making its previous parent a leaf node. In turn, this new leaf node may perform the same process to find that the attack is happening further up the chain.
- The leaf node is unable to find a new parent, possibly because there are no other nodes in range. In this case, the node may notify its parent that it is unable to find a new parent, and request that its parent perform HCR further up the chain. This may be done with a specialized control message (by setting a free flag - still to be worked out in full).

In the first scenario, the coordinated blackhole attack is successfully mitigated, always resulting in better network performance than before the mitigation (as any packet delivery is better than no delivery at all). In the second scenario, it might be possible that the parent node is in fact the one performing the blackhole attack - if this is the case, no change in network performance will be achieved. If the parent node is not an attacking node however, HCR can recursively be applied by each node's parent until the attack is either mitigated, or a node with no alternative parents is reached. A node will be able to choose a new parent in most cases, therefore HCR should have decent mitigation performance. As HCR uses DAO messages to periodically check for root reachability, there is some Control Message Overhead (CMO) to be taken into account. With aggressive monitoring of 5 root-checks in every 20 seconds, CMO may rise by up to 25%. However, lower monitoring rates of around 10 root-checks per 10 minutes should still significantly improve network performance in case of an attack, while only raising CMO by 1.6%.

5.2 Importance of addressing coordinated attacks

RPL is known as a low-security protocol. The constrained nature of LLNs makes it impossible to make use of security methods used in the regular internet. Many attacks are able to disrupt the routing topology, packet delivery rate, latency and other metrics of an RPL network, even with only one attacker. In practice however, most attacks with only a single attacker already have proposed detection and/or mitigation methods or may only have a small impact on the total network. When multiple attackers start colluding though, impact of the attack may grow greatly and proposed detection methods for single attackers may become completely useless (think of the Watchdog method against the coordinated blackhole attack). Therefore, it is important that research is done not only to mitigate single attackers, but also to mitigate colluding attackers.

5.3 Importance of addressing the coordinated blackhole attack

The coordinated blackhole attack is a specifically nasty attack, which is able to disconnect large sections of a DODAG from the rest of the network. A coordinated blackhole attack in the first hop from the root node may essentially take out all, or half of the network. It is essential that this attack, which leads to a drop of all packets received, is mitigated in a performant fashion.

5.4 Trade-offs and limitations of HCR

Theoretically, HCR performs well against the coordinated blackhole attack whenever nodes have the option to switch parents. This is because the ability to switch parents is what allows the attacked nodes to regain root node connectivity. According to the analysis in section 3, HCR has 100% detection rate and poses varying mitigation performance based on the number of alternative parents.This mitigation method does not work when nodes are unable to switch parents. In such a scenario the attacked node is able to identify that it is under attack, but will not be able to do anything about it. In practice, this situation should rarely occur and if it does, there is no way to defend against it either way, as the only way to inform other nodes of the attack is through the misbehaving parent node. HCR is somewhat limited in its scalability, as CMO scales linearly with the number of leaf nodes in a network. This performance could be improved by allowing intermediate nodes to perform their own root check and returning an intermediate ACK. Note that it is important this ACK may not be spoofed by an attacking node. There are also some trade-offs to be taken into consideration with HCR: the ACK threshold, and the frequency of root-checking are the main trade-offs. A higher threshold means more ACKs may be missed before a node will look for a new parent. This means more DAO messages need to be sent to the root node, resulting in a higher CMO. A lower threshold will result in a lower CMO, but may run into issues when packets are dropped due to lossy links or other incidental performance drops. The other trade-off is the frequency of root-checking. This frequency determines the interval between root-checks, and may include sleep time between intervals to prevent CMO from rising too much.

Unfortunately, this version of HCR does not work if the malicious nodes are only dropping data packets, while forwarding control messages as usual. This problem can be avoided by using data packets aimed for the root node. In such a scenario it is important that the root node recognizes that it must send a reply packet. It is also important that the

attacking node is unable to distinguish these special packets from regular data packets. As the assumption is made that the blackhole method is dropping all packets, this issue is out of scope of this paper and may be addressed in future work.

5.5 Future research

Some future research directions in the field of RPL-based IoT networks are mentioned here. This research is aimed to help understanding, and validation of other research in this field.

- *Research into the impact of collusion attacks against RPL-based IoT networks.* A survey on which collusion attacks exist, and what their respective impact on the performance of RPL is would be greatly beneficial for coming up with comparisons on performance overhead of newly proposed mitigating methods. This would require an extensive amount of work, as most colluding attacks do not have research posed yet. Therefore this could be split up into separate research for each colluding attack that can be found.
- *Research into possible link-layer, or application layer security against colluding attacks against RPL-based IoT networks.* It may be possible that link-layer security may already be able to detect when an attack is being performed, thereby reducing the need for routing layer security protocols, which may induce more overhead.
- *Research into control messages disguised as data packets.* This research is aimed to be an extension of this paper. As mentioned earlier, HCR is unable to function if the blackhole nodes are only dropping data packets. To this end, it would be beneficial if control messages could be sent as indistinguishable data packets.

6 Conclusion

To answer the research question: *Identify the state-of-the-art detection and mitigation solutions proposed specifically for coordinated routing attacks in RPL-based IoT networks. What limitations do these solutions exhibit? What are possible approaches that could be used to address one or more of these limitations?*, a study on colluding attacks has been performed. The coordinated blackhole attack was discussed in this paper, as it is detrimental for RPL networks' performance and does not have a dynamic mitigating solution. To address this issue, HCR is presented, which is a mitigation technique using a periodic pinging of the root node. Leaf nodes periodically send a DAO message towards the root node to ensure that they are able to reach it. They check for a response by setting the 'K' flag in the DAO message, after which the root node will return a DAO-ACK upon reception. When too many ACKs are missed, the affected node will look for a new parent. Depending on the parameters chosen for HCR, CMO may rise anywhere between 1.6% and 25%, with a 100% detection rate. Mitigation rate depends on the structure of the network, varying anywhere between 0% in the (rare) worst case, up to 100% in the best case. The mitigation rate and parameter optimization are to be addressed in future work.

References

[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials*, 17(4):2347–2376, 2015.

[2] Ahmet Arış, Sıddıka Berna Örs Yalçın, and Sema F. Oktuğ. New lightweight mitigation techniques for RPL version number attacks. *Ad Hoc Networks*, 85:81–91, 2019.

[3] Ismail Butun, Patrik Osterberg, and Houbing Song. Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. *IEEE Communications Surveys and Tutorials*, 22(1):616–644, 2020.

[4] Bin Fan, Dave G. Andersen, Michael Kaminsky, and Michael D. Mitzenmacher. Cuckoo Filter. pages 75–88, 2014.

[5] Faraz Idris Khan, Taeshik Shon, Taekkyeun Lee, and Kihyung Kim. Wormhole attack prevention mechanism for RPL based LLN network. *International Conference on Ubiquitous and Future Networks, ICUFN*, pages 149–154, 2013.

[6] Hyung Sin Kim, Jeonggil Ko, David E. Culler, and Jeongyeup Paek. Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey. *IEEE Communications Surveys and Tutorials*, 19(4):2502–2525, 2017.

[7] J Tripathi, J De Oliveira, and J P Vasseur. Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL). pages 1–40, 2012.

[8] Luc Vinet and Alexei Zhedanov. A 'missing' family of classical orthogonal polynomials. Technical Report 8, 2011.

[9] Tianchen Zhang, Taimin Zhang, Xiaoyu Ji, and Wenyuan Xu. Cuckoo-RPL: Cuckoo filter based RPL for defending AMI Network from blackhole attacks. *Chinese Control Conference, CCC*, 2019-July:8920–8925, 2019.